

部 門	競 技 部 門	No.1 登録番号	30004
-----	---------	-----------	-------

No.2	1) 予定開発期間：5 ヶ月							
	2) 予定開発人数：3 人(ソルバー:2 人、GUI:1 人)							
		4月	5月	6月	7月	8月	9月	10月
	問題分析		←————→					
	設計			←————→				
実装				←————→				
試用・トレーニング					←————→			

No.3	実現方法																																																																							
	<p>1) 原画像の推察アルゴリズム</p> <p>現時点では断片画像の配置を全探索することを考えている。この方法の計算回数は階乗オーダーになりそうだが、適切に枝刈りをすれば、探索空間はおそらく制限時間内に十分探索可能な広さになると考えられる。具体的には、各断片画像が隣り合った時の「不自然さ」という量を導入し、この量の最大値があらかじめ定めておいた上限を超えた場合に、その状態以降の探索を打ち切るというような方法を考えている。さらに、分割統治的に探索を行う（ブロック単位で枝刈りを行う）ことで、より少ない計算回数で原画像の推察を行うことができると予想される。</p> <p>隣り合った断片画像に対する「不自然さ」は二つの画像の境界部分における色相や彩度、明度等から計算する。原画像が十分予想可能であるという前提に基づくと、原画像における断片画像同士の境界部分では色の変化が小さいはずであるため、この評価方法は妥当だと考えている。</p>																																																																							
	<p>2) 断片画像の並び替えアルゴリズム</p> <p>ビームサーチなどを用いることを考えている。盤面の初期状態やサイズからコストの上限が定まるため、その上限を超えた変形は選ばないようにするといった工夫ができると思われる。</p> <p>また、断片画像が十分少ない場合は、各状態（盤面と選択した断片画像の組）をノードとして最短経路問題に帰着させることで、最適解を求めることができるため、断片画像の数によって並び替えアルゴリズムを変えることも考えている。</p> <p style="text-align: center;">断片画像が少ない場合の探索例</p> <div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="text-align: center;"> <p>マス目</p> <table border="1" style="border-collapse: collapse;"> <tr><td>5</td><td>2</td><td>4</td></tr> <tr><td>3</td><td>7</td><td>6</td></tr> <tr><td>1</td><td>8</td><td>9</td></tr> </table> <p>選択</p> </div> <div style="text-align: center;"> <p>→ A</p> <table border="1" style="border-collapse: collapse;"> <tr><td>5</td><td>2</td><td>4</td></tr> <tr><td>3</td><td>7</td><td>6</td></tr> <tr><td>1</td><td>8</td><td>9</td></tr> </table> <p>1</p> </div> <div style="text-align: center;"> <p>→ B</p> <table border="1" style="border-collapse: collapse;"> <tr><td>5</td><td>7</td><td>4</td></tr> <tr><td>3</td><td>6</td><td>2</td></tr> <tr><td>1</td><td>8</td><td>9</td></tr> </table> <p>2</p> </div> <div style="text-align: center;"> <p>→ A</p> <table border="1" style="border-collapse: collapse;"> <tr><td>5</td><td>7</td><td>4</td></tr> <tr><td>3</td><td>2</td><td>6</td></tr> <tr><td>1</td><td>8</td><td>9</td></tr> </table> <p>3</p> </div> <div style="text-align: center;"> <p>→ B</p> <table border="1" style="border-collapse: collapse;"> <tr><td>5</td><td>4</td><td>2</td></tr> <tr><td>3</td><td>7</td><td>6</td></tr> <tr><td>1</td><td>8</td><td>9</td></tr> </table> <p>2</p> </div> <div style="text-align: center;"> <p>→ B</p> <table border="1" style="border-collapse: collapse;"> <tr><td>2</td><td>5</td><td>4</td></tr> <tr><td>3</td><td>7</td><td>6</td></tr> <tr><td>1</td><td>8</td><td>9</td></tr> </table> <p>2</p> </div> <div style="text-align: center;"> <p>→ B</p> <table border="1" style="border-collapse: collapse;"> <tr><td>3</td><td>5</td><td>4</td></tr> <tr><td>2</td><td>7</td><td>6</td></tr> <tr><td>1</td><td>8</td><td>9</td></tr> </table> <p>2</p> </div> <div style="text-align: center;"> <p>→ B</p> <table border="1" style="border-collapse: collapse;"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td>4</td><td>5</td><td>6</td></tr> <tr><td>7</td><td>8</td><td>9</td></tr> </table> <p>X</p> </div> </div> <p style="text-align: right;">選択コスト:A 交換コスト:B</p> <p style="text-align: right;">最終的な状態</p> <p style="text-align: right;">※Xは1から9の整数</p>	5	2	4	3	7	6	1	8	9	5	2	4	3	7	6	1	8	9	5	7	4	3	6	2	1	8	9	5	7	4	3	2	6	1	8	9	5	4	2	3	7	6	1	8	9	2	5	4	3	7	6	1	8	9	3	5	4	2	7	6	1	8	9	1	2	3	4	5	6	7	8
5	2	4																																																																						
3	7	6																																																																						
1	8	9																																																																						
5	2	4																																																																						
3	7	6																																																																						
1	8	9																																																																						
5	7	4																																																																						
3	6	2																																																																						
1	8	9																																																																						
5	7	4																																																																						
3	2	6																																																																						
1	8	9																																																																						
5	4	2																																																																						
3	7	6																																																																						
1	8	9																																																																						
2	5	4																																																																						
3	7	6																																																																						
1	8	9																																																																						
3	5	4																																																																						
2	7	6																																																																						
1	8	9																																																																						
1	2	3																																																																						
4	5	6																																																																						
7	8	9																																																																						

No.4	3) その他（独創的なところ）
	<p>GUI とソルバーの相互開発において、アジャイル開発を取り入れることにより、仕様変更に対応することができる。継続的インテグレーションを取り入れ、バグの早期発見によるリードタイムの短縮、工数の削減を図る。原画像の推定の際に境界だけを見ることにより計算量を減らしている。1 台の PC をサーバとし、残り 2 台の PC を接続させ、分散処理を行う。</p>

No.4	開発環境 IDE:VSCode, Visual Studio Language:C++, C# FrameWork:STD, boost GitHub
------	---